

# SPECIFICATION

Electronic Version 1.2.8

Stylesheet Version 1.0

## [MESSAGE TRANSMITTING QUEUE AND ASSOCIATED METHOD]

### Cross Reference to Related Applications

This application claims the priority benefit of Taiwan application serial no. 91100090, filed January 7, 2002.

### Background of Invention

[0001] Field of Invention

[0002] The present invention generally relates to a queue, and more particularly, to a message transmitting queue and the method of operating the same.

[0003] Description of Related Art

[0004] In the system with dual processors, in order to achieve the objective of transmitting a message from one to the other, a message transmitting queue is commonly used as a bridge in between. However, if the message transmitting queue, for example, uses the general serial queue that only has the write pointer and the read pointer, although this may be okay for the single task system, a message misplacement or overlap may occur in the multi-threaded system. Since the size of the transmitted message between the processors may exceed the atomic data read/write size of the processor, the message is actually transmitted to the queue by several times. Moreover, the write pointer cannot move until the message writing completes. Therefore, in the multi-threaded system, when a process does not complete writing the message, the other process may complete writing the message at this time point in advance. So, the previous message is overlapped. When the write

pointer moves, the message that does not complete the writing operation is then written by the previous process, thus causing message misplacement. Such a circumstance can be prevented by using the semaphore. However, this will introduce a complicated software operation and impact the system performance.

## Summary of Invention

[0005] Therefore, the present invention provides a message transmitting queue and the method for operating the same, applied to a multi-threaded system to prevent the misplacement or the overlap of data and command in the transmitted message without using the software semaphore.

[0006] The present invention provides a message transmitting queue to provide an access media for transmitting messages between the source controller and the destination controller, comprising a plurality of message rows, a write control unit and a read control unit. The plurality of message rows is used to store the message that the source controller intends to transmit to the destination controller. Moreover, each message row at least comprises a write complete flag and a distribution complete flag. The write control unit is coupled to the source controller and the plurality of message rows, wherein when the source controller intends to write the message, the write control unit sequentially outputs the address of the free message row according to the distribution complete flag. When the source controller completes reading the address of the message row, the write control unit sets the distribution complete flag of the message row. When the source controller completes writing the message of the message row, the write control unit sets the write complete flag of the message row, and when the message transmitting queue does not have a free message row, outputs a no free message row signal. The read control unit is coupled to the destination controller and the plurality of message rows. When the write complete flag of the message row that is sequentially read is set, the read control unit issues a read request to inform the destination controller to read the message of the message row, and when the destination controller completes reading the message, clears the distribution complete flag and the write complete flag of the message row.

[0007] In the preferred embodiment of the present invention, the write control unit comprises a write pointer control unit, a distribution complete flag multiplexer, and a

distribution address multiplexer. The write pointer control unit is used to store a write address of the message row of the message transmitting queue. When the source controller completes reading the write address, the write pointer control unit sets the distribution complete flag of the message row pointed to by the write address and progresses the write address, and when the source controller completes writing the message of the message row, the write pointer control unit sets the write complete flag of the message row. The distribution complete flag multiplexer, that is coupled to the write pointer control unit and the plurality of distribution complete flags of the plurality of message rows, is used to output a not-distributed signal according to the distribution complete flag of the message row pointed to by the write address. The distribution address multiplexer, that is coupled to the distribution complete flag multiplexer and the write pointer control unit, is used to determine and output either the write address or the no free message row signal according to the not-distributed signal.

[0008] Furthermore, the read control unit comprises a read pointer control unit, a read buffer, and a read request multiplexer. The read pointer control unit is used to store a read address of the message transmitting queue, wherein when the destination controller completes reading the message of the message row pointed to by the read address, the read pointer control unit clears the distribution complete flag and the write complete flag of the message row and progresses the read address. The read buffer that is coupled to the read pointer control unit and the plurality of message rows is used to output the message of the message row pointed to by the read address. The read request multiplexer, that is coupled to the read pointer control unit and the plurality of write complete flags of the plurality of message rows, is used to output the read request according to the write complete flag of the message row pointed to by the read address.

[0009] In the preferred embodiment of the present invention, the size of the message that the source controller intends to transmit to the destination controller is two times of the atomic data read/write size of the source controller. For example, each message comprises data and command, and the size of data and command is the atomic data read/write size of the source controller. It is preferred that the size of the data row and the command row is a multiple of four bytes.

[0010] The present invention further provides an operating method of the message transmitting queue for transmitting the message of the source controller to the destination controller. The message transmitting queue comprises a plurality of message rows, a write pointer and a read pointer. Moreover, each message row at least comprises a write complete flag and a distribution complete flag. The method comprises the steps of: at first, setting the write pointer and the read pointer to point to the first message row address; when the source controller completes reading the write pointer, setting the distribution complete flag of the message row pointed to by the write pointer and progressing to the write pointer; when the source controller completes writing the message of the message row, setting the write complete flag of the message row; when the write complete flag of the message row pointed to by the read pointer is set, issuing a read request; and when the destination controller completes reading the message of the message row pointed to by the read pointer, clearing the distribution complete flag and the write complete flag of the message row pointed to by the read pointer and progressing the read pointer.

[0011] In the preferred embodiment of the present invention, when the write pointer progresses and points to a message row whose distribution complete flag is set, a no free message row signal is issued to inform the source controller. Wherein, the read request is the interrupt request of the central processing unit.

[0012] As shown in the description above, the message transmitting system that applies the message transmitting queue of the present invention allows the source controller individually to write the message such as data and command according to the message row address distributed by the message transmitting queue. Moreover, the destination controller reads the message sequentially according to the write complete flag and the read address. The system operated according to this mechanism prevents the message misplacement and overlap that may happen in the traditional queue, and does not have to use the complicated semaphore that incurs a software burden. Therefore, the system performance is improved significantly.

## Brief Description of Drawings

[0013] The accompanying drawings are included to provide a further understanding of the invention, and are incorporated in and constitute a part of this specification. The

drawings illustrate embodiments of the invention, and together with the description, serve to explain the principles of the invention. In the drawings,

[0014] FIG. 1 schematically shows a block diagram of the message transmitting queue according to one preferred embodiment of the present invention; and

[0015] FIG. 2 schematically shows an operating flow chart of the message transmitting queue according to one preferred embodiment of the present invention.

## Detailed Description

[0016] FIG. 1 schematically shows a block diagram of the message transmitting queue of the preferred embodiment according to the present invention. As shown in the diagram, the message transmitting queue 100 is used as the access media for transmitting messages between the source controller 110 and the destination controller 120. The source controller 110 and the destination controller 120 are preferably central processing units, comprising a plurality of message rows 130, a write control unit 165, and a read control unit 195. For clarity, the present invention only uses four of the message rows 130 to explain herein. However, as known to those in the art, the number of the message rows can be extended flexibly to the number that is required. These four message rows 130 are used to store the message that the source controller 110 intends to transmit to the destination controller 120. For clarity, it is assumed that the address is 00, 01, 10 and 11 from top down respectively. Moreover, each message row 130 shown in the diagram respectively comprises the write complete flag I0, I1, I2 and I3, the distribution complete flag C0, C1, C2 and C3, the command rows SrcOutIRQCmd0, SrcOutIRQCmd1, SrcOutIRQCmd2 and SrcOutIRQCmd3 that are used to store the command that the source controller 110 intends to transmit to the destination controller 120, and the data rows SrcOutIRQMsg0, SrcOutIRQMsg1, SrcOutIRQMsg2 and SrcOutIRQMsg3 that are used to store the data that the source controller 110 intends to transmit to the destination controller 120. The size of the command row and the data row is preferably four bytes or a multiple of four bytes depending on the requirement of the message that is transmitted.

[0017]

The write control unit 165 comprises a write pointer control unit 140, a

distribution complete flag multiplexer 150, and the distribution address multiplexer 160. The write pointer control unit 140 is used to store a write address of the message row 130 of the message transmitting queue 100, and when the source controller 110 completes reading the write address, sets the distribution complete flag of the message row 130 pointed to by the write address and progresses the write address. When the source controller 110 completes writing the message of the message row 130, the write pointer control unit sets the write complete flags of the message rows 130. The distribution complete flag multiplexer 150 that is coupled to the write pointer control unit 140 and the distribution complete flags of the message rows 130, outputs a write address or a not-distributed signal according to the distribution complete flags of the message rows 130. The distribution address multiplexer 160 that is coupled to the distribution complete flag multiplexer 150 and the write pointer control unit 140, determines to output either the write address or the no free message row signal according to the not-distributed signal.

[0018] The read control unit 195 comprises a read pointer control unit 170, a read buffer 180, and the read request multiplexer 190. The read pointer control unit 170 is used to store the read address of the message transmitting queue 100. When the destination controller 120 completes reading the message of the message row 130 pointed by the read address, clears the distribution complete flag and the write complete flag of the message row 130 are cleared and the read address is increased. The read buffer 180 that is coupled to the read pointer control unit 170 and the message row 130, is used to output the message of the message row 130 pointed to by the read address. The read request multiplexer 190 that is coupled to the read pointer control unit 170 and the write complete flags of the message rows 130, is used to output the read request according to the write complete flag of the message row 130 pointed to by the read address. The read request can be an interrupt request of the central processing unit.

[0019] In this embodiment, when the system is reset, the distribution complete flags C0~C3 and the write complete flags I0~I3 are all reset to 0, the write address stored in the write pointer control unit 140 is 00, and the read address stored in the read control unit 170 is 00. In other words, both the write address and the read address point to the message row of address 00. At this moment, the first process of the

source controller 110 demands to transmit a message to the destination controller 120, and the first process returns the currently free address 00 of the message row in the write pointer control unit 140 to the source controller 110 via the distribution complete flag multiplexer 150 and the distribution address multiplexer 160. When the reading completes, the write pointer control unit 140 sets the distribution complete flag C0 to 1, and progresses the write address to 1 to point to the next free message row address 01. At this moment, if the second, the third and the fourth process demand to transmit messages, the free message row addresses 01, 10, 11 are read in sequence respectively, the corresponding distribution complete flags C1, C2, C3 are set to 1, and the write address is progressed back to 00 to point to the first message row that has been distributed. If another process demands to transmit a message before any write message has been read out, the process finds out that there is no free message row via the distribution complete flag multiplexer 150 and the distribution address multiplexer 160. Therefore, there is no free message row until the message occupying the first message row is read out to release its space. In one case, the second process has completed writing data and command to the message row at address 01 prior to the first process. At this moment, the write pointer control unit 140 sets the write complete flag I1 to 1. Since the read address at the read pointer control unit 170 is still 00, the read request multiplexer 190 does not issue the read request until the first process completes writing data and command to the message row at address 00 and sets the write complete flag I0 to 1. When the read request multiplexer 190 issues the read request, the destination controller 120 reads the message indicated by read address 00 from the read buffer 180. The read pointer control unit 170 clears the distribution complete flag C0 and the write complete flag I0 and progresses the read address to 01 after the reading has completed. At this moment, since the write complete flag I1 of the message row at address 01 pointed by the read address had been set due to the fact that the second process has written the message in advance. The read request multiplexer 190 issues another read request, so the destination controller 120 continues to read the message at address 01. Then, the read address points to the message row at address 02. The message at address 02 cannot be read until the source controller 110 has written the message and set the write complete flag I2. Therefore, the message can be transmitted efficiently without message misplacement or overlapping.

[0020] In other words, the write control unit 165 intends to constantly point to a message row that is currently free. However, whether it is really in a free state is determined according to the distribution complete flag of the currently free message row to prevent any previous message that has not been processed yet from being overlapped. If it is determined that it is really in the free state, the address of the currently free message row is returned to the source controller 110. The returned address of the message row is preferably stored by the firmware till the related message has been written to the message row that is currently free, to prevent message misplacement. If it is determined that the currently free message row is not in the free state, a no free message row signal is returned to the source controller 110. The source controller 110 will issue a request to ask for the free message row from the message transmitting queue 100 after a predetermined period of time.

[0021] More particularly, the distribution address multiplexer 160 sequentially reads the free message row addresses 00, 01, 10, 11 generated by the write control unit 165 (or the write pointer) in this embodiment. The four bytes size, the atomic read/write size that can be processed by the controller, is exemplified for description hereafter. The address 00 mentioned above may correspond to the physical address 0x0C00, 0x0C04, and the address 01 mentioned above may correspond to the physical address 0x0C08, 0x0C0C, etc. A no free message row signal is returned to the source controller 110, for example, by returning 0xFFFF to the source controller 110, which falls in an unreasonable addressing range to inform the source controller 110 that there is no free message row available now.

[0022] As described above, when the source controller 110 intends to write the message, the write control unit 165 sequentially outputs the address of the message row 130 that is free according to the distribution complete flags. Moreover, when the source controller 110 completes reading the currently free address of the message row 130, the write control unit 165 sets the distribution complete flag of the message row 130. When the source controller 110 completes writing the message of the message row 130, the write control unit 165 sets the write complete flag of the message row 130. When the message transmitting queue 100 does not have any free message row 130, the write control unit 165 outputs a no free message row signal, such as 0xFFFF, to instruct the source controller 110 to wait for a short period of time, so that the queue



that is busy can first complete part of its job, and its own resource can be utilized during this waiting period. The read control unit 195 issues a read request to inform the destination controller 120 to read the message of the message row 130 when the write complete flag of the message row 130 that is read sequentially is set. The distribution complete flag and the write complete flag of the message row 130 are both cleared after the destination controller 120 completes reading the message.

[0023] A message transmitting queue is used to transmit a plurality of messages from the source controller to the destination controller, comprising a plurality of message rows, a write pointer and a read pointer. Moreover, each message row at least comprises a write complete flag and a distribution complete flag. The operating method is shown in FIG. 2, comprising the steps of: in step S210, when the system is reset, setting the write pointer and the read pointer to point to the first message row address; in step S220, when the source controller completes reading the write pointer, setting the distribution complete flag of the message row pointed to by the write pointer and progressing the write pointer; in step S230, when the source controller completes writing the message into the message row, setting the write complete flag of the message row; in step S240, when the write complete flag of the message row pointed to the read pointer is set, issuing a read request; and in step S250, when the destination controller completes reading the message from the message row pointed to by the read pointer, clearing the distribution complete flag and the write complete flag of the message row pointed to the read pointer and progressing the read pointer. When the write pointer progresses and points to a message row whose distribution complete flag is set, a no free message row signal is output to inform the source controller that there is no free message row to be used anymore.

[0024] In summary, the present invention discloses a message transmitting queue that is coupled to the source controller and the destination controller and is used to transmit messages between the source controller and the destination controller. The message transmitting queue comprises a plurality of message rows, a write control unit and a read control unit. A plurality of message rows are used to store the messages, and each message row at least comprises a write complete flag and a distribution complete flag. The write control unit having the write pointer is coupled to the source controller and the message rows, and sequentially outputs the addresses of the free

message rows to the source controller to store the messages according to the distribution complete flags and the write pointer. The read control unit having the read pointer is coupled to the destination controller and the message rows, and sequentially issues the read requests, for example, interrupt requests to inform the destination controller to read the message according to the write complete flags and the read pointer. When the length of each message exceeds the atomic read/write size that can be processed by the controllers, the write control unit determines whether a free message row exists according to the write pointer and the distribution complete flag of the message row pointed to by the write pointer. If the write control unit determines that a free message row exists, the message row address pointed to by the write pointer is returned to the source controller, the distribution complete flag of the message row pointed to by the write pointer is set, and the write pointer is moved to the next message row. Otherwise, a no free message row signal is returned to the source controller. If the source controller receives the no free message row signal, the source controller issues a request to ask for a free message row from the message transmitting queue after waiting for a predetermined period of time. When the destination controller completes reading a message row, the read control unit clears the distribution complete flag and the write complete flag of the message row pointed to by the read pointer, and moves the read pointer to the next message row. Every time the source controller receives the address of a free message row, the received free message row address is stored by the firmware till the associated message has been written into the free message row.

[0025] Therefore, the message transmitting system that applies the message transmitting queue of the present invention allows the source controller individually to write the message according to the message row address distributed by the message transmitting queue. Moreover, the destination controller reads the message sequentially according to the write complete flag and the read address. The system according to this invention prevents message misplacement and overlap that may happen in the traditional queue, and does not have to use the complicated semaphore that incurs software burden. Therefore, the system performance is improved significantly.

[0026] Although the invention has been described with reference to the above

embodiment thereof, it will be apparent to one of the ordinary skill in the art that modifications to the described embodiment may be made without departing from the spirit of the invention. Accordingly, the scope of the invention will be defined by the attached claims not by the above detailed description.